Circuit visiting 10 ordered vertices in infinite grids

David Coudert, Frédéric Giroire, Ignasi Sau

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Circuit visiting 10 ordered vertices in infinite grids

David Coudert — Frédéric Giroire — Ignasi Sau

N° 6910

Avril 2009

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--6910--FR+ENG

Circuit visiting 10 ordered vertices in infinite grids

David Coudert*†, Frédéric Giroire*† , Ignasi Sau*†

Thème COM — Systèmes communicants
Projets Mascotte

Rapport de recherche n° 6910 — Avril 2009 — 19 pages

Abstract:

A *circuit* in a simple undirected graph $G = (V, E)$ is a sequence of vertices $\{v_1, v_2, \ldots, v_{k+1}\}$ such that $v_1 = v_{k+1}$ and $\{v_i, v_{i+1}\} \in E$ for $i = 1, \ldots, k$. A circuit C is said to be *edge-simple* if no edge of G is used twice in C. In this article we study the following problem: which is the largest integer k such that, given any subset of k ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the k vertices in the prescribed order? We prove that $k = 10$. To this end, we first provide a counterexample implying that $k < 11$. To show that $k \geq 10$, we introduce a methodology, based on the notion of core graph, to reduce drastically the number of possible vertex configurations, and then we test each one of the resulting configurations with an ILP solver.

Key-words: Connectivity, simple circuit, grid.

This work was partially funded by Région PACA, European project IST FET AEOLUS, Ministerio de Ciencia e Innovación, European Regional Development Fund under project MTM2008-06620-C03-01/MTM, and Catalan Research Council under project 2005SGR00256.

* MASCOTTE, INRIA, I3S, CNRS, Univ. Nice Sophia, Sophia Antipolis, France.
† firstname.lastname@sophia.inria.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Circuit visitant 10 sommets ordonnés dans la grille infinie

Résumé : Un circuit dans un graphe non-orienté $G = (V, E)$ est une séquence de sommets $\{v_1, v_2, \ldots, v_{k+1}\}$ telle que $v_1 = v_{k+1}$ et $\{v_i, v_{i+1}\} \in E$ pour tout $i = 1, \ldots, k$. Un circuit C est simple par rapport aux arêtes si aucune arête n'est utilisée 2 fois. Dans ce rapport, nous étudions le problème suivant: quelle est le plus grand entier K tel que, pour tout sous-ensemble de k sommets ordonnés dans la grille infinie, il existe un circuit simple par rapport aux arêtes visitant les sommets dans l'ordre prescrit ? Nous prouvons que $k = 10$. Pour obtenir ce résultat, nous donnons un contre-exemple impliquant que $k < 11$. Puis, pour montrer que $k \geq 10$, nous introduisons la notion de *core graph* pour réduire de façon drastique le nombre possible de configurations de sommets, et nous testons chacune de ces configurations avec un solveur de programmes linéaires en nombres entiers.

Mots-clés : Connectivité, circuit simple, grille.

1 Introduction

A *circuit* in a simple undirected graph $G = (V, E)$ is a sequence of vertices $\{v_1, v_2, \ldots, v_{k+1}\}$ such that $v_1 = v_{k+1}$ and $\{v_i, v_{i+i}\} \in E$ for $i = 1, \ldots, k$. A circuit C is said to be *edge-simple* if no edge of G is used twice in C. The existence of a circuit through a prescribed set of vertices or edges has been an important graph-theoretical question for many years [6, 19, 12, 15, 4, 3, 8, 7, 16, 2, 11, 10]. Typically, high connectivity is a powerful sufficient condition for the existence of such circuits. For instance, it is well known that in a k-vertex-connected graph any subset of k nodes [6] or any subset of $k-1$ independent edges [12] is included in a cycle. A circuit C is a *cycle* if no vertex of G is used twice in C, except for $v_1 = v_{k+1}$.

However, knowing specific properties of the graph often permits to prove much stronger results. In this article we focus on the existence of edge-simple circuits through specified vertices in the infinite square grid (or equivalently, a large enough torus), which is a widely studied 4-connected graph. In addition, we do not require the circuit only to visit a subset of vertices, but also to visit them in a prescribed order. It is clear that such a circuit in the square grid always exists for any ordered subset of 4 vertices. After thinking for a few minutes it is also easy to convince oneself that the same holds for 5 vertices. On the other hand, it seems intuitive to suspect that this property will not be true for an arbitrary large subset of ordered vertices of the square grid. Therefore, the following question arises: which is the largest integer k such that, given *any* subset of k ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the k vertices in the prescribed order? Here, we prove that $k = 10$.

To obtain this result, one has a priori to test the existence of an edge-simple circuit visiting k vertices in the prescribed order on the grid, for all possible placements and orderings of the k vertices. Since the number of possible placements and orderings is prohibitively large, we introduce a methodology, based on the notion of *core graph*, to reduce the number of configurations to be tested. We first provide some background and motivations for the problem in Section 2. We then show in Section 3 that checking the feasibility of a configuration on the grid is equivalent to checking its feasibility on an auxiliary graph, called *internal graph*. Then, in Section 4 we introduce the notion of *core graphs* to reduce drastically the number of internal graphs to be tested. In Section 5 we give a counterexample establishing the upper bound $k \leq 10$. In Section 6 we match this upper bound with an ILP solver to exhaustively test all the orderings for a small list of possible configurations that we obtained after applying the reductions of Sections 3 and 4. Finally, Section 7 concludes the article.

2 Background and Motivation

Connectivity is one of the cornerstone concepts of graph theory. Maybe the most archetypal results are Menger's classical theorems [5], which say that a graph is k-vertex-connected (resp. k-edge-connected) if and only if it contains k vertex-disjoint (resp. edge-disjoint) paths between any two vertices. There is a huge literature concerning extremal problems of cycles in k-connected graphs. For instance, it is well known that in a k-vertex-connected

graph any subset of k nodes [6] or any subset of $k-1$ independent edges [12] is included in a cycle. There are a number of works giving necessary or sufficient conditions for the existence of a cycle through a specified set of vertices in a general graph [15, 4, 7, 16].

Some stronger results have been given for specific classes of graphs, like 3-connected cubic graphs [8, 7]. For this class of graphs it is known that there exists a cycle through any 9 vertices, and that there exists a cycle which passes through any 10 given vertices if and only if the graph is not contractible to the Petersen graph [8] in such a way that each of the 10 vertices maps to a distinct vertex of the Petersen graph. If, in addition, the 3-connected cubic graph is planar, then there exists a cycle through any 23 vertices [2]. Another example can be found in [10], where the authors provide necessary and sufficient conditions for a given graph embedded on the torus to contain edge-disjoint cycles satisfying prescribed topological properties.

The disjoint paths problem. Observe that, in a general (di)graph, the problem of deciding whether there exist edge-disjoint paths between given pairs of vertices is NP-complete [14] (even if the graph is a square grid [17]). When the number of pairs of vertices is bounded by a constant, the disjoint paths problem is polynomial in undirected graphs [20], NP-complete in directed graphs [18] (even with only two pairs of vertices [9]), and polynomial in symmetric directed graphs [13].

However, all these results do not take into account the *order* in which the cycle visits the prescribed set of nodes. This is a natural constraint, since for example in telecommunication networks it may be important to connect a subset of nodes in such a way that each node numbered i has capability to communicate only with the two nodes numbered $i-1$ and $i+1$ (modulo the cardinality of the subset of nodes). This could be the case, for instance, of the classical *token ring* networks defined by the standard *IEEE* 802.5. That is, there exists a whole class of problems to consider when the constraint on the order is introduced. In this article we study one of these problems in square grids.

When designing a telecommunication network, the fault tolerance is a crucial issue. Observe that the simplest network which is able to support any single link failure is an edge-simple circuit, and that is one of the main reasons why the study of such circuits is important. The study of the square grid is also natural, due among other reasons to its extensive use in parallel computing. In this context, it is interesting to know which is the largest integer k for which there always exists a circuit visiting any ordered subset of at most k nodes. Observe also that without taking into account the ordering, there exists a cycle (and thus, a circuit) visiting any subset of vertices of the square grid, since the square grid is a Hamiltonian graph.

It is worth mentioning that the square grid is in some sense the common *skeleton* of planar graphs. Indeed it is well-known that every planar graph of branchwidth at least ℓ contains an $(\lfloor \ell/4 \rfloor \times \lfloor \ell/4 \rfloor)$-grid as a minor [21]. Therefore, a square grid is *inside* every planar graph, and any edge-disjoint circuit in a minor of a graph can be easily transformed to an edge-disjoint circuit in the graph itself.

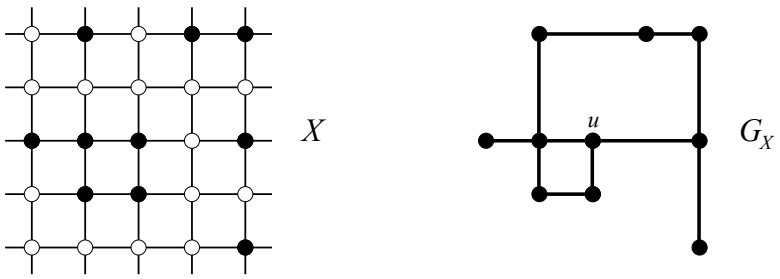


Figure 1: A configuration X (defined by the full dots) and its corresponding internal graph G_X.

3 Preliminaries

In this section we introduce some definitions to be used throughout. We use standard graph terminology (see, for instance, [5]).

Definition 1 (Internal graph, internal and external degree)**.** *Given a subset X of nodes in the square grid, the* internal graph $G_X = (V, E)$ *of X is the graph with $V = X$, and for $u, v \in X$, $\{u, v\} \in E$ if and only if u and v are on the same row (or column) and there is no other $z \in X$ between u and v on that row (or column).*

Given $u \in X$, the internal degree $d_{in}(u)$ *of u is the degree of u in the internal graph G_X of X, i.e., $d_{G_X}(u)$. Similarly, the* external degree *of $u \in X$ is $d_{out}(u) = 4 - d_{in}(u)$. A vertex $u \in X$ is* isolated *if $d_{in}(u) = 0$.*

For example, in Fig. 1, a configuration X in the square grid (defined by the full dots) and its corresponding internal graph G_X are depicted. The vertex labeled u satisfies $d_{\text{in}}(u) = 3$ and $d_{\text{out}}(u) = 1$.

Since we deal with an infinite square grid, any two vertices of an internal graph G_X with external degree at least one can be connected with a path in the grid without using any edge of G_X. This is because a vertex that has external degree at least one has no neighbor in the internal graph along an infinite semirow or semicolumn of the grid. This fact can be modeled in the following way: given an internal graph G_X, we construct a graph $\hat{G}_X$ from G_X by adding a new vertex ∞ and, for each vertex $u \in V(G_X)$, $d_{\text{out}}(u)$ copies of the edge $\{u, \infty\}$.

Definition 2 (Feasible)**.** *An internal graph G_X is* feasible *if, for all the permutations σ of the vertices of G_X, there exists an edge-simple circuit joining the vertices of G_X following the ordering given by σ. A configuration X is* feasible *if G_X is feasible.*

Observe that the fact that G_X is feasible is a sufficient (but *not* necessary) condition for the existence of an edge-simple circuit visiting the vertices of X. Intuitively, the internal

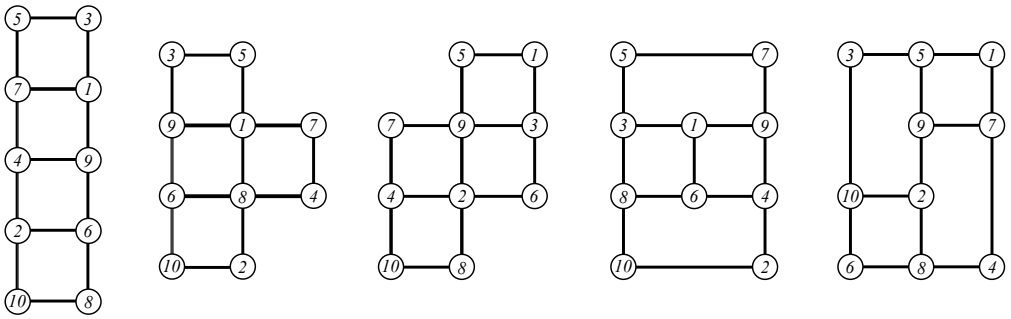


Figure 2: Some feasible internal graphs on 10 vertices.

graph captures the most difficult case among all the configurations having the same internal graph. By the above discussion, a configuration X is feasible if and only if there exists an edge-simple circuit in $\hat{G}_X$ visiting the vertices of G_X in the prescribed order.

Before getting into technical results, and in order to get familiar with the problem, the curious reader may verify that the internal graphs on 10 vertices depicted in Fig. 7 (together with a numbering of their vertices) accept a solution. We shall see in Section 6 that this fact is not a coincidence, since any configuration on 10 vertices is feasible.

4 Reducing the Problem

We now prove several technical lemmata to be used in the sequel of the article. The objective is to reduce the number of configurations to be tested.

Lemma 1. *Any internal graph in which all vertices have external degree at least* 2 *is feasible.*

Proof. Let G be an internal graph in which all vertices have external degree at least 2, and assume that the vertices are ordered $v_1, v_2, \ldots, v_k$ by the permutation σ. Then the circuit $\{v_1, \infty, v_2, \infty, v_3, \ldots, v_{k-1}, \infty, v_k, \infty, v_1\}$ is a solution in G. □

Lemma 2. *If an internal graph G is feasible then any subgraph $H \subseteq G$ is feasible.*

Proof. Let G be a feasible internal graph, and let H be a subgraph of G. Assume first that $V(H) = V(G)$, and let $v_1, \ldots, v_k$ be an ordering of the vertices of H. Consider a solution C in G for the same ordering $v_1, \ldots, v_k$ of the vertices of G. A solution in H is obtained from C by replacing each each $\{u, v\} \in E(G) \setminus E(H)$ with the edges $\{u, \infty\}, \{\infty, v\}$. Otherwise, if $|V(H)| = k < n = |V(G)|$, given an ordering $v_1, \ldots, v_k$ of $V(H)$, consider a solution C in G for an ordering of $V(G)$ that coincides with $v_1, \ldots, v_k$ when restricted to $V(H)$. Then the above replacement transforms C into a solution in H. □

Two internal graphs G_1 and G_2 are said to be *equivalent* if G_2 is feasible if and only if G_1 is.

Lemma 3. *Any two isomorphic internal graphs G_1 and G_2 are equivalent.*

Proof. Let φ be an isomorphism mapping the vertices of G_1 to the vertices of G_2, and suppose w.l.o.g. that G_1 is feasible. We shall find a solution in G_2 for an arbitrary ordering $v_1, \ldots, v_k$ of its vertices. Let C be a solution in G_1 for the ordering $\varphi^{-1}(v_1), \ldots, \varphi^{-1}(v_k)$, which exists since G_1 is feasible. Let $P_{i,j}$ be the subpath of C going from $\varphi^{-1}(v_i)$ to $\varphi^{-1}(v_j)$. Since each path P_1 in G_1 corresponds bijectively to a path P_2 in G_2, we can denote without ambiguity the path P_2 by $\varphi(P_1)$. Then the circuit defined by the union of the paths $\varphi(P_{1,2}), \varphi(P_{2,3}), \ldots, \varphi(P_{k-1,k}), \varphi(P_{k,1})$ is a solution in G_2. □

Lemma 4. *If an internal graph G_X is feasible, then so is the internal graph $G_{X'}$ obtained from G_X via the following transformation T_1:*

(1) Choose from G_X an isolated vertex u and an edge $\{x, y\}$.

(2) Remove u, add a new vertex v, and replace the edge $\{x, y\}$ with the edges $\{x, v\}, \{v, y\}$.

Proof. Given an ordering σ of the vertices of $G_{X'}$, let s (resp. t) be the vertex that precedes (resp. follows) v in σ. Consider a solution C in $\hat{G}_X$ for the ordering obtained from σ by replacing v with u, and let us obtain from C a solution in $\hat{G}_{X'}$. If the edge $\{x, y\}$ is used in C, replace it with the path $\{x, v, y\}$. Since $d_{G_X}(u) = 0$, the path from s to u in $\hat{G}_X$ is of the form $\{s, P, \infty, u\}$, and the path from u to t is of the form $\{u, \infty, Q, t\}$. Since $d_{G_{X'}}(v) = 2$, replacing the path $\{s, P, \infty, u\}$ (resp. $\{u, \infty, Q, t\}$) with $\{s, P, \infty, v\}$ (resp. $\{v, \infty, Q, t\}$) yields a solution in $\hat{G}_{X'}$. □

Lemma 5. *If an internal graph G_X is feasible, then so is the internal graph $G_{X'}$ obtained from G_X via the following transformation T_2:*

(1) Choose from G_X two vertices u and w, such that u is isolated and $d_{in}(w) \leq 3$.

(2) Remove u, and add a new vertex v and the edge $\{w, v\}$.

Proof. Given an ordering σ of the vertices of $G_{X'}$, let s (resp. t) be the vertex that precedes (resp. follows) v in σ. Consider a solution C in $\hat{G}_X$ for the ordering obtained from σ by replacing v with u, and let us obtain from C a solution in $\hat{G}_{X'}$. If any of the edges $\{w, \infty\}$ is used in C, replace one of them with the edges $\{w, v\}, \{v, \infty\}$. Since $d_{G_X}(u) = 0$, the path from s to u in $\hat{G}_X$ is of the form $\{s, P, \infty, u\}$, and the path from u to t is of the form $\{u, \infty, Q, t\}$. Since $d_{G_{X'}}(v) = 1$, replacing the path $\{s, P, \infty, u\}$ (resp. $\{u, \infty, Q, t\}$) with $\{s, P, \infty, v\}$ (resp. $\{v, \infty, Q, t\}$) yields a solution in $\hat{G}_{X'}$. □

Combining inductively Lemmas 4 and 5, we deduce that if $G_{X'}$ is an internal graph obtained from a feasible graph G_X with a sequence of the transformations T_1 and T_2, then $G_{X'}$ is also feasible. In practice, this means that in any internal graph we can take the vertices that lie in the *middle* of a path and the vertices with internal degree one, and put them as isolated vertices. If the resulting configuration is feasible, then by Lemmas 4 and 5, so is the original one. In other words, we can restrict ourselves to internal graphs G_X whose

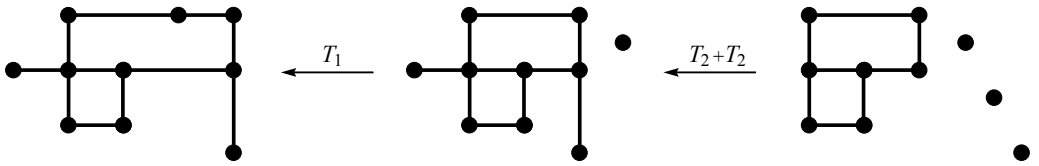


Figure 3: We can restrict ourselves to core graphs. An arrow from a graph G to a graph H means that if G is feasible, so is H (due to either transformation T_1 or transformation T_2).

connected components (except isolated vertices) have at least two vertices in each row and each column.

Definition 3 (Core graph, ℓ-core graph)**.** *An internal graph is a* core graph *if all its non-edgeless connected components have at least two vertices in each row and each column. A core graph G on k vertices is an* ℓ-core graph *if G has $k - \ell$ isolated vertices.*

Lemmas 4 and 5 imply that we can restrict ourselves to core graphs. For instance, consider the example of Fig. 3. The leftmost internal graph (which is the same example of Fig. 1) can be obtained by a sequence of the transformations T_1 and T_2. Thus, to prove that the three internal graphs of Fig. 3 are feasible it is enough to prove it for the rightmost graph, which is a 7-core graph.

This simplification reduces the number of configurations dramatically. In particular, the above discussion together with Lemma 1 proves that all forests are feasible. Therefore, if we want to know if all the configurations on k vertices are feasible, it suffices to test all the core graphs on k vertices; this is the topic of Section 6 for $k = 10$. Summarizing,

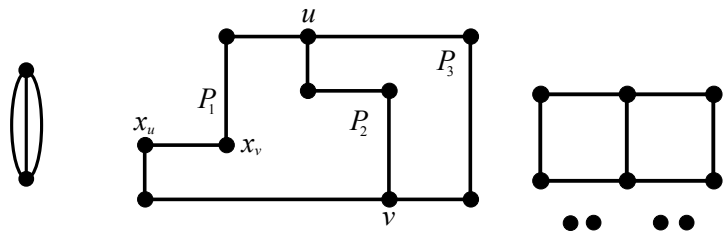
Proposition 6. *If all the core graphs on k vertices are feasible, then all the configurations on k vertices are feasible.*

Note that if all the configurations on k vertices are feasible, then clearly so are all the configurations on k' vertices, for every $k' < k$.

We introduce a last criterium to deduce the feasibility of an internal graph on 10 vertices.

Lemma 7. *All the 10-core graphs on 10 vertices whose non-edgeless connected components can be obtained from a triple edge by subdividing edges are feasible.*

Proof. Let G be a 10-core graph whose non-edgeless connected components can be obtained from a triple edge (see Fig. 4(a)). It is easy to check that the smallest graph in the square grid that can be obtained in this way has 6 vertices (see Fig. 4(c)), so since G has 10 vertices, G has exactly one such non-edgeless component. It is also not difficult to check that all the vertices of G have internal degree 2, except isolated vertices and 2 vertices u and v that have internal degree 3, as it is exemplified in Fig. 4(b). Suppose we are given an arbitrary ordering $x_1, \ldots, u, \ldots, v, \ldots, x_{10}$ of the vertices of G. By an argument analogous to that given in the



(a) (b) (c)

Figure 4: **(a)** a triple edge, **(b)** a 10-core obtained from subdivision, and **(c)** the only 6-core.

proof of Lemma 1, to prove the feasibility of G it is sufficient to find an internal path (i.e., using only edges of G) from u and v to one of their adjacent vertices in the ordering. We distinguish three cases according to the relative position of u and v in the ordering.

If u and v are consecutive in the ordering, any path from u to v in G fulfills the conditions. If u and v are at distance 2 in the ordering, let w be the vertex which lies between u and v. Then take a path in G from u to v through w. Otherwise, if u and v are at distance at least 3, u (resp. v) has at least one vertex at distance 1 in the ordering x_u (resp. x_v) inside G. Observe that there are 3 edge-simple paths P_1, P_2, P_3 between u and v in G. Suppose that $x_u \in P_i$ and $x_v \in P_j$. If $i \neq j$, take a path from u to x_u in P_i and a path from v to x_v in P_j. If $i = j$, assume w.l.o.g. that $i = j = 1$. If, beginning from u, x_u comes before x_v in P_1, consider the subpaths in P_1 from u to x_u and from v to x_v. Otherwise, if x_v comes before x_u in P_1 (see Fig. 4(b)), consider the paths $\{u, P_2, v \xrightarrow{\text{in } P_1} x_u\}$ and $\{v, P_3, u \xrightarrow{\text{in } P_1} x_v\}$. □

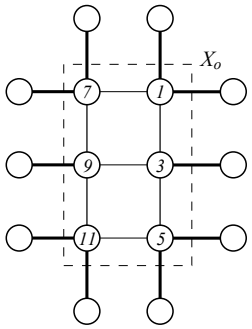


Figure 5: Counterexample showing that $k < 12$.

5 Upper Bound: $k < 11$

In this section we show an unfeasible counterexample proving that $k < 11$. For the sake of the presentation, we first describe a simple configuration showing that $k < 12$.

Given a set $X = \{1, 2, 3, \ldots, 12\}$ of ordered nodes in a square grid G, let X_e (resp. X_o) be the subset of nodes of X with an even (resp. odd) number, and note that any path joining two consecutive vertices must go from X_o to X_e, or viceversa. Let X_o be the set displayed in Fig. 5. Then, regardless of the placement of X_e, we need at least 12 edges outgoing from the graph induced by X_o to route the 12 paths, but there are only 10 such edges (the thick edges in Fig. 5). So, this configuration is unfeasible for any placement of X_e.

Before providing the counterexample showing that $k < 11$, we need the following definition.

Definition 4 (Internal path)**.** *Given an internal graph G_X, a permutation σ of X, a solution C to the instance (G_X, σ), and a subset $S \subseteq X$, an* internal path in S *is a subpath P of C linking two consecutive vertices of X according to σ, such that P is a subgraph of $G_X[S]$.*

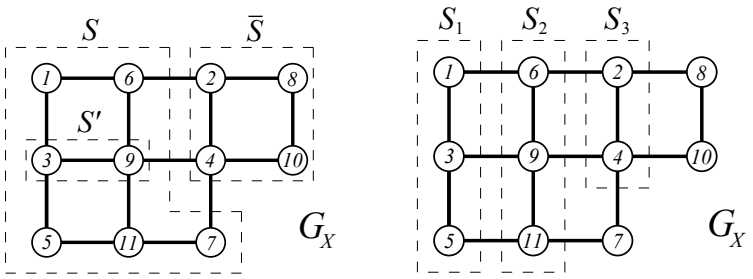
Given a subset of vertices $S \subseteq X$, the *paths originating from* S are paths with at least one endpoint in S.

Proposition 8. $k < 11$.

Let (G_X, σ) be the internal graph on 11 vertices together with the ordering depicted in Fig. 6.

Proof. Suppose for the sake of contradiction that there exists a solution C to the instance (G_X, σ). Let $S = \{1, 6, 3, 9, 5, 11, 7\} \subseteq X$, and let $\overline{S} = \{2, 8, 4, 10\}$ (see Fig. 6(a)). Note that there are 12 edges outgoing from $G_X[S]$ to the rest of the grid.

Claim 1. *C contains exactly 1 internal path in S.*



(a) (b)

Figure 6: Counterexample (G_X, σ) of Proposition 8 showing that $k < 11$, together with the vertex sets defined in the proof.

Proof. Suppose first that there is no internal path in S. Therefore, each path originating from S uses at least 2 edges outgoing from $G_X[S]$. Since $|S| = 7$, there must be 14 edges in C outgoing from $G_X[S]$ to the rest of the grid, but there are only 12. Therefore, C contains at least 1 internal path in S.

Suppose now that C contains at least 2 internal paths in S. Let $S' = \{3, 9\}$ (see Fig. 6(a)), and note that there are 6 edges outgoing from $G_X[S']$. Note also that the only possible internal paths in S are $5 \to 6$, $6 \to 7$, and $11 \to 1$, so any internal path in S must cross S'. Therefore, there can be at most 2 such internal paths, and those 2 paths use 4 edges outgoing from $G_X[S']$. Thus, only $6 - 4 = 2$ outgoing edges from $G_X[S']$ are left, which are not enough to route the 4 subpaths in C containing the vertices of S'. Therefore, C contains exactly 1 internal path in S. □

Claim 1 implies all the edges outgoing from $G_X[S]$ are used by C to route paths originating at S. Let $S_1 = \{1, 3, 5\}$ and $S_3 = \{2, 4\}$ (see Fig. 6(b)).

Claim 2. *C contains at least 2 internal paths from S_1 to S_3.*

Proof. Note that subgraph $G_X[S_3]$ has 6 outgoing edges. Since all the edges outgoing from $G_X[S]$ are used by C, exactly 3 paths go from S to $\overline{S}$ in C. Clearly, the 4 subpaths in C containing the vertices of S_3 use 4 outgoing edges from $G_X[S_3]$. Note that all paths from S to S_3 are from S_1.

If there is no path in C from S_1 to S_3, then the 3 paths from S to $\overline{S}$ cross S_3, so no edge outgoing from $G_X[S_3]$ would be left to route the paths originating from S_3, which is a contradiction.

If there is 1 path in C from S_1 to S_3, then 2 paths from S to $\overline{S}$ cross S_3, so altogether the 3 paths from S to $\overline{S}$ use 5 out of the 6 outgoing edges from $G_X[S_3]$. However, 3 additional outgoing edges from $G_X[S_3]$ would be needed to route the 3 remaining paths originating from S_3, which is a contradiction. □

Consider now $S_2 = \{6, 9, 11\}$ (see Fig. 6(b)). The subgraph $G_X[S_2]$ has 8 outgoing edges, and 6 of them are required in C to route the paths originating at S_2, so only 2 edges outgoing from $G_X[S_2]$ are still available in C. But, by Claim 2, C contains at least 2 internal paths from S_1 to S_3 (which cross S_2), hence using 4 outgoing edges from $G_X[S_2]$. The proposition follows. □

6 Lower Bound: $k \geq 10$

To show that $k \geq 10$, one has a priori to test all the configurations with 10 vertices on the grid are feasible. But, the number of such configurations is prohibitively big, as testing a single configuration may take a non-negligible (see discussion below). Hence we introduce a methodology, based on the notion of *core graph* (see the results of Section 4), to reduce the number of configurations to be tested.

A naïve strategy to generate all configurations is to consider all the possibilities of placing 10 points in the square grid. However, we showed in Proposition 6 that we only need to consider *core graphs* with 10 vertices, introduced in Definition 3. In addition, these core graphs can be considered modulo isomorphism, see Lemma 3. It is clear that the smallest integer i such that an i-core on 10 vertices exists is 4, and in that case the non-edgeless connected component of the 4-core is a 4-cycle. Such a core is always feasible due to Lemma 1, because all the vertices have external degree at least 2. It is also easy to check that, due to the topology of the grid, a 5-core cannot exist. One can also verify that the only 6-core in which not all vertices have external degree at least 2 is a 2×3-grid (see Fig. 4(c)). Therefore, it is enough to test this 6-core plus all the ℓ-cores on 10 vertices, for $\ell = 7, 8, 9, 10$. The procedure to generate the core graphs to be tested is detailed in Algorithm 1. The complete code and some examples as well can be found at [1].

Proposition 9. *The feasibility of any configuration on 10 vertices follows from Algorithm 1.*

Proof. To prove the correctness of Algorithm 1, we analyze each step sequentially.

Steps 1-2: As by definition, core graphs have at least 2 vertices in each row (resp. column) (besides isolated vertices), the number of rows (resp. column) of an l-core is at most $\lfloor \frac{\ell}{2} \rfloor$. Hence, it is enough to consider internal graphs fitting into an $\left(\lfloor \frac{\ell}{2} \rfloor \times \lfloor \frac{\ell}{2} \rfloor\right)$-grid, except possibly a set of isolated vertices. Moreover, two internal graphs G_1 and G_2 such that G_2 is obtained from G_1 by a translation or a symmetry are clearly equivalent.

In steps 3-6, the algorithm removes from $\mathcal{T}_\ell$ all the internal graphs with some vertex of internal degree 1. This can be done because, by Lemma 5, it is enough to test the internal graphs in which all vertices, except the isolated ones, have internal degree at least 2.

In steps 7-10, the algorithm removes from $\mathcal{T}_\ell$ all the internal graphs such that all vertices have external degree at least 2, which are feasible due to Lemma 1.

In steps 14-15, the algorithm removes from $\mathcal{T}_\ell$ all the isomorphism classes that have some representant without at least 2 vertices per row and column. This can be done by combining Lemma 3 and Proposition 6.

Algorithm 1 Test configurations on 10 vertices

for $\ell = 6$ to 10 **do**
  generate a list $\mathcal{T}_\ell$ of all the internal graphs on ℓ vertices in an $(\lfloor \frac{\ell}{2} \rfloor \times \lfloor \frac{\ell}{2} \rfloor)$-grid (modulo translations, symmetries, and compression of empty rows or columns)
  // `INTERNAL DEGREE 1:`
  for each $G \in \mathcal{T}_\ell$ such that G has some vertex of internal degree 1 **do**
    remove G from $\mathcal{T}_\ell$ {*Lemma 5*}
  // `EXTERNAL DEGREE AT LEAST 2:`
  for each $G \in \mathcal{T}_\ell$ such that all the vertices of G have external degree at least 2 **do**
    remove G from $\mathcal{T}_\ell$ {*Lemma 1*}
  // `ISOMORPHIC GRAPHS:`
  partition $\mathcal{T}_\ell$ into classes $\mathcal{G}_1, \ldots, \mathcal{G}_n$ of isomorphic graphs
  for $i = 1$ to n **do**
    if there exists $G \in \mathcal{G}_i$ such that G does *not* have at least two vertices per row and column **then**
      remove from $\mathcal{T}_\ell$ all the graphs in $\mathcal{G}_i$ {*Lemma 3 and Proposition 6*}
    else
      remove from $\mathcal{T}_\ell$ all the graphs in $\mathcal{G}_i$ except one {*Lemma 3*}
  // `SUBDIVISION OF TRIPLE EDGE:`
  if $\ell = 10$ **then**
    for each $G \in \mathcal{T}_\ell$ such that G can be obtained from a triple edge by subdividing edges **do**
      remove G from $\mathcal{T}_\ell$ {*Lemma 7*}
  // `SUBGRAPHS:`
  for each pair of graphs $G, H \in \mathcal{T}_\ell$ such that H is a subgraph of G **do**
    remove H from $\mathcal{T}_\ell$ {*Lemma 2*}
  $b_\ell \leftarrow 1$
  for each $G \in \mathcal{T}_\ell$ **do**
    $G' \leftarrow G + (10 - \ell)$ isolated vertices
    for each permutation σ of the vertices of G' **do**
      test if (G', σ) is feasible using an LP solver
      if (G', σ) is *not* feasible **then**
        $b_\ell \leftarrow 0$
if $(b_6 \cdot b_7 \cdot b_8 \cdot b_9 \cdot b_{10}) == 1$ **then**
  $k = 10$
else
  $k < 10$

By Lemma 3, all the graphs in an isomorphism class are equivalent, so it is enough to keep one graphs of each class, as it is done in steps 16-18.

The correctness of steps 20-25 (resp. 26-29) follows directly from Lemma 7 (resp. Lemma 2).

In steps 31-39, the feasibility of each pair (G', σ) is tested using an ILP solver to find a solution of the associated integer multicommodity flow problem, as explained in Section 2 (more details about the implementation are available at [1]).

Finally, by Proposition 6, it is clear that $k = 10$ if and only if all the ℓ-core graphs are feasible, for each $\ell \in \{6, 7, 8, 9, 10\}$. □

Remark. In step 12 of Algorithm 1, we partition T_ℓ into isomorphism classes. This step could take a non-negligible time if we just test if each pair of graphs are isomorphic. To deal with this problem, we first carry out a sieve according to the sorted degree sequence of the vertices and the sorted degree sequences of the neighbours of each vertex. That is, if two graphs do not have the same sequence of degrees and degrees of the neighbours of each vertex, we infer directly that these two graphs are not isomorphic. This sieve reduces the computation time considerably.

Remark. Observe that, due to Lemma 4, the internal graphs without at least 2 vertices per row and column could have been already removed from T_ℓ after step 2. The reason why we kept those graphs until step 15 is that some graphs that do have at least 2 vertices per row and column are isomorphic to graphs without at least 2 vertices per row and column, so we can also remove them from T_ℓ.

Table 1 summarizes the number of ℓ-cores obtained while running Algorithm 1, for $\ell \in \{6, 7, 8, 9, 10\}$. The numbers given in the first row (initial number of internal graphs) follow from the introduction of internal graphs; without it, we would have a much greater number of configurations to test. Note that the results of Section 4 induce an overall reduction from 4714 to 52 graphs.

Testing the feasibility of core graphs. Recall that for each core graph G on 10 vertices, G is feasible if for any ordering of $V(G)$ there is an edge-simple circuit visiting $V(G)$ in the prescribed order. W.l.o.g. we can assign to one of the vertices of G the number 1 of the permutation (modulo cyclic permutations), and then for each core graph one has to test $9! = 362.880$ possibilities.

For each core graph G and permutation σ, the problem we study can be easily formulated as an integer multicommodity flow problem in a graph with unitary capacity on the edges and so as an integer linear program (ILP). Indeed, the existence of an edge-simple circuit C_σ in a core graphs G is equivalent to the existence of k edge-disjoint paths in G between the pairs of vertices (or *commodities*) $\{\sigma(1), \sigma(2)\}, \ldots, \{\sigma(k-1), \sigma(k)\}, \{\sigma(k), \sigma(1)\}$. Thus, a feasible solution of the ILP implies the existence of an edge simple circuit, and this feasibility can be quickly checked using an ILP solver (for instance, CPLEX).

In average, testing the 9! permutations for each internal graph takes around 40 minutes on a PC with an Intel Core 2 Duo CPU 2.33GHz running Fedora 8 (see [1]), so testing the

Graphs \ ℓ	6	7	8	9	10	Total
Initial number of internal graphs	1	7	53	485	4166	4714
Number of isomorphisms	0	3	42	453	4051	4581
Number of subgraphs	0	0	5	10	58	73
Number of single graphs	0	2	6	22	74	104
Final number of internal graphs	1	2	4	10	35	52

Table 1: Number of ℓ-core graphs on 10 vertices in Algorithm 1. A *single* graph is a graph with a line or column with only one vertex.

4714 internal graphs would take around 4 months and a half. Testing the 52 remaining graphs has taken just 35 hours and 37 minutes [1].

Running the ILP solver on the configurations given by Algorithm 1, we obtained that all ℓ-cores are feasible for each $\ell \in \{6, 7, 8, 9, 10\}$. Therefore, combining Propositions 8 and 9 yields that

Theorem 10. *There exists an edge-simple circuit through any set of 10 ordered vertices of an infinite square grid.*

7 Concluding Remarks

In this article we showed that given any subset of 10 ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the 10 vertices in the prescribed order, and that the number 10 cannot be replaced by 11. To do so, we introduced a methodology to reduce the problem to a small number of configurations, which were then exhaustively tested using an ILP solver. The details about the implementation of our algorithm are available at [1]. Finding a purely combinatorial proof of this result remains open.

Another avenue for further research could be to impose a bound on the size of the grid or torus, namely to consider an $(n_1 \times n_2)$-torus and to find the largest integer $k(n_1, n_2)$ such that given any subset of $k(n_1, n_2)$ ordered vertices in an $(n_1 \times n_2)$-torus, there exists an edge-simple circuit visiting the $k(n_1, n_2)$ vertices in the prescribed order.

Another direction is to consider another graphs instead of the square grid, like triangular and hexagonal grids and, more generally, general planar graphs or graphs of bounded treewidth.

Finally, adding the constraint of the prescribed order to the classical problems concerning the existence of circuits (see related work in Section 2), creates a whole family of new problems to consider.

References

[1] `http://www-sop.inria.fr/members/Frederic.Giroire/circuit`.

[2] R. Aldred, S. Bau, D. Holton, and B. McKay. Cycles through 23 vertices in 3-connected cubic planar graphs. *Graphs and Combinatorics*, 15:373–376, 1999.

[3] S. Bau and D. Holton. Cycles containing 12 vertices in 3-connected cubic graphs. *J. Graph Theory*, 15(4):421–429, 1991.

[4] J. Bondy and L. Lovász. Cycles through specified vertices of a graph. *Combinatorica*, 1:117–140, 1981.

[5] R. Diestel. *Graph Theory*. Springer-Verlag, 2005.

[6] G. Dirac. In abstrakten Graphen vorhandene vollständige 4-Graphen u nd ihre Unterteilungen. *Math. Nachr.*, 22:61–85, 1960.

[7] Y. Egawa, R. Glas, and S. Locke. Cycles and paths through specified vertices in k-connected graphs. *J. Comb. Theory Ser. B*, 52:20–29, 1991.

[8] M. Ellingham, D. Holton, and C. Little. Cycles through ten vertices in 3-connected cubic graphs. *Combinatorica*, 4:265–273, 1984.

[9] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Journal of Theoretical Computer Science*, 10(2):111–121, 1980.

[10] A. Frank and A. Schrijver. Edge-Disjoint Circuits in Graphs on the Torus. *J. Comb. Theory Ser. B*, 55(1):9–17, 1992.

[11] F. Göringa, J. Harant, E. Hexel, and Z. Tuzac. On short cycles through prescribed vertices of a graph. *Discrete Mathematics*, 286(1-2):67–74, 2004.

[12] R. Häggkvist and C. Thomassen. Circuits through specified edges. *Discrete Math.*, 41:29–34, 1982.

[13] A. Jarry and S. Pérennes. Disjoint paths in symmetric digraphs. *Discrete Applied Mathematics*, 157(1):90–97, 2009.

[14] R. M. Karp. On the complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[15] K. Kawarabayashi. Cycles through a prescribed vertex set in n-connected graphs. *J. Comb. Theory Ser. B*, 90(2):315–323, 2004.

[16] A. Kelmans and M. Lomonosov. When m vertices in a k-connected graph cannot be walked round along a simple cycle. *Discrete Math.*, 38:317–322, 1982.

[17] M. Kramer and J. van Leeuwen. Wire-routing is NP-complete. Technical Report RUU-CS-82-4, Department of Computer Science, University of Utrecht, 1982.

[18] A. S. LaPaugh and R. L. Rivest. The subgraph homeomorphism problem. *J. Comput. Syst. Sci.*, 20(2):133–149, 1980.

[19] L. Lovász. On some connectivity properties of Eulerian graphs. *Acta Math. Acad. Sci. Hungar.*, 28:129–138, 1976.

[20] N. Robertson and P. Seymour. Graph minors XIII: the disjoint paths problem. *J. Comb. Theory Ser. B*, 63:65–110, 1995.

[21] N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Comb. Theory Ser. B*, 62(2):323–348, 1994.

A A Necessary Condition for the Existence of a Circuit

The *bisection width* of a graph G, commonly denoted by $bw(G)$, is the minimum number of edges of all cuts disconnecting the graph into two components whose number of vertices differs on at most 1. Using this concept we can prove a result that states a simple necessary condition for the existence of an edge-simple circuit on an internal graph with even number of vertices.

Proposition 11. *If an internal graph $G = (V, E)$ is feasible, with $|V|$ even, then*

$$bw(G) \geq |E| - |V|. \tag{1}$$

Proof. Let $|V| = n$ and $|E| = m$. For each permutation of the vertices of G, finding an edge-simple circuit in G is equivalent to find n edge-disjoint paths joining the ordered vertices of G. Let P_{out} be the maximum number of paths that can be routed using some external edge, and equivalently let P_{in} be the number of paths that can be routed using *only* internal edges. It is clear that, if a cycle exists, then $n \leq P_{in} + P_{out}$. But P_{out} is at most half of the sum of all external degrees, that is half of $\sum_{v \in G}(4 - d_G(v))$. On the other hand, we have that $P_{in} \leq bw(G)$. The last inequality can be easily deduced by assigning the odd numbers of the permutation to the vertices of one of the components of the bisection, and the even numbers to the vertices of the other component. Thus,

$$n \leq P_{in} + P_{out} \leq bw(G) + \frac{\sum_{v \in G}(4 - d_G(v))}{2} = bw(G) + 2n - m.$$

From the preceding equation we get the result. □

Proposition 11 gives an easy but powerful way to prove that a graph is not feasible. For instance, to deduce Proposition 8 we have just to place X_e symmetrically next to X_o. Then $bw(G) = 3 < |E| - |V| = 17 - 12 = 5$. Therefore G is not a feasible graph, so $k < 12$. Proposition 11 motivates the following natural definition.

Definition 5. *An unlabeled configuration is said to be* critical *if its internal graph satisfies Equation (1) with equality.*

For instance, it is easy to check that all the configurations with 10 vertices depicted in Fig. 7 are critical, regardless of the numbering (these configurations are more symmetrical that those on 11 vertices). Nevertheless, the curious reader can verify that all of them are feasible (this may take some time if one is not familiar with the problem...). We have proved in Section 6 that this fact is not a coincidence.

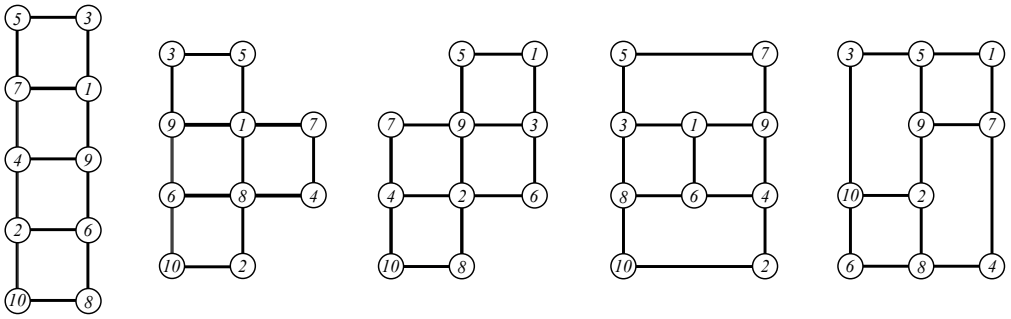


Figure 7: Some critical configurations with 10 vertices.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399